On Coloring Resilient Graphs

Jeremy Kun Lev Reyzin

Department of Mathematics, Statistics, and Computer Science
University of Illinois at Chicago
Chicago, IL 60607
{jkun2,lreyzin}@math.uic.edu

Abstract

We introduce a new notion of resilience for constraint satisfaction problems, with the goal of more precisely determining the boundary between NP-hardness and the existence of efficient algorithms for resilient instances. In particular, we study r-resiliently k-colorable graphs, which are those k-colorable graphs that remain k-colorable even after the addition of any r new edges. We prove lower bounds on the NP-hardness of coloring resiliently colorable graphs, and provide an algorithm that colors sufficiently resilient graphs. This notion of resilience suggests an array of open questions for graph colorability and other combinatorial problems.

1 Introduction and related work

An important goal in studying NP-complete combinatorial problems is to find precise boundaries between tractability and NP-hardness. This is often done by adding constraints to the instances being considered until a polynomial time algorithm is found. For instance, while SAT is NP-hard, the restricted 2-SAT and XOR-SAT versions are decidable in polynomial time.

In this paper we present a new angle for pushing the boundary of NP-hardness. We informally define the resilience of a constraint-based combinatorial problem and we focus on the case of resilient graph colorability. Roughly speaking, a positive instance is resilient if it remains a positive instance up to the addition of a constraint. For example, an instance G of Hamiltonian circuit would be "r-resilient" if G has a Hamiltonian circuit, and G minus any r edges *still* has a Hamiltonian circuit. In the case of coloring, we say a graph G is r-resiliently k-colorable if G is k-colorable and will remain so even if any r edges are added. One would imagine that *finding* an r-coloring in a very resilient positive instance would become easier, as that instance is very "far" from being not colorable. And in general, one can pose the question: how resilient can instances be and have the search problem still remain hard?[1]

For most NP-hard problems, there are natural definitions of resiliency in our framework. For instance, resilient positive instances for optimization problems over graphs can be defined as those that remain positive instances even up to the addition or removal of any edge. For satisfiability, we say a resilient instance is one where variables can be "fixed" and the formula remains satisfiable. In problems like set-cover, we could allow for the removal of a given number of sets. Indeed, this can be seen as a general notion of resilience applying to fixing constraints in constraint satisfaction problems (CSPs), which have an extensive literature [28].

In this paper, however, we focus on graph coloring. Resilience is defined up to the addition of edges, and we first show that this is an interesting notion: many famous, well studied graphs exhibit strong resilience properties. Then, perhaps surprisingly, we prove that 3-coloring a 1-resiliently 3-colorable graph is NP-hard – that is, it is hard to color a graph even when it is guaranteed to remain 3-colorable under the addition of

[1] We focus on the search versions of the problems because the decision version on resilient instances induces the trivial "yes" answer.

any edge. Briefly, our reduction works by mapping positive instances of 3-SAT to 1-resiliently 3-colorable graphs and negative instances to graphs of chromatic number at least 4. An algorithm which can color 1-resiliently 3-colorable graphs can hence distinguish between the two. On the other hand, we observe that 3-resiliently 3-colorable graphs have polynomial-time coloring algorithms (leaving the case of 3-coloring 2-resiliently 3-colorable graphs tantalizingly open). We also show that efficient algorithms exist for k-coloring $\binom{k}{2}$-resiliently k-colorable graphs for all k, and discuss the implications of our lower bounds.

This paper is organized as follows. In the next two subsections we review the literature on other notions of resilience and on graph coloring. In Section 2 we inspect a resilient version of 6-SAT which lays the groundwork for our main theorem on 1-resilient 3-coloring. In Section 3 we formally define the resilient graph coloring problem and present preliminary upper and lower bounds. In Section 4 we prove our main theorem, and in Section 5 we discuss open problems.

1.1 Related work on resilience

There are related concepts of resilience in the literature. Perhaps the closest in spirit is Bilu and Linial's notion of stability [6]. Their notion is restricted to problems over metric spaces and attempts to explain why real-world instances are easy. They argue that practical instances often exhibit some degree of stability, which can make the problem easier. Their results on clustering stable instances have seen considerable interest and have been substantially extended and improved [4, 6, 31]. Their study is also not limited to clustering – for instance one can study TSP and other optimization problems over metrics under the Bilu-Linial assumption [30]. A related notion of stability by Ackerman and Ben-David [1] for clustering yields efficient algorithms when the data lies in Euclidian space.

Our notion of resilience, on the other hand, is most natural in the case when the optimization problem has natural constraints, which can be fixed or modified. Our primary goal is also different – we seek to more finely delineate the boundary between tractability and hardness in a systematic way across problems.

Property testing can also be viewed as involving resilience. Roughly speaking property testers are randomized algorithms which distinguish between combinatorial structures that satisfy a property or are very far from satisfying it. These algorithms are typically given access to a small sample depending on ε alone (and not the size of the input). For graph property testing, as with resilience, the concept of being "far" from having a property involves requiring the addition or removal of an arbitrary set of at most $\varepsilon\binom{n}{2}$ edges from G. Our notion of resilience is different in that we consider adding or removing a constant number of constraints. More importantly, property testing is more concerned with query complexity than with computational hardness, whereas we seek to relate the complexity of search problems across varying degrees of resilience.

1.2 Previous work on coloring

As the technical results of this paper are on graph colorability, we review the relevant work.

A graph G is k-colorable if there is an assignment of k distinct colors to the vertices of G so that no edge is monochromatic. Determining whether G is k-colorable is a classic an NP-hard problem [22]. Many attempts to simplify the problem, such as assuming planarity or bounded degree of the graph in question, still result in NP-hardness [11]. A large body of work surrounds positive and negative results for explicit families of graphs. The list of families that are polynomial-time colorable includes triangle-free planar graphs, perfect graphs and almost-perfect graphs, bounded tree- and clique-width graphs, quadtrees, and various families of graphs defined by the lack of an induced subgraph [18, 26, 13, 9, 27].

With little progress on coloring general graphs, research has naturally turned to approximation. In approximating the chromatic number of a general graph, the first results were of Garey and Johnson, giving a performance guarantee of $O(n/\log n)$ colors [20] and proving that it is NP-hard to approximate chromatic number to within a constant factor less than two [14]. Further work improved this bound by logarithmic factors [5, 16]. In terms of lower bounds, Zuckerman [33] derandomized the PCP-based results of Håstad [17] to prove the best known approximability lower-bound to date, $O(n^{1-\varepsilon})$.

There has been much recent interest in coloring graphs which are already known to be colorable while minimizing the number of colors used. For a 3-colorable graph, Widgerson gave an algorithm using at most $O(n^{1/2})$ colors [32], which Blum improved to $\tilde{O}(n^{3/8})$ [7]. A line of research based on semidefinite programming and other combinatorial improvements improved this bound still further to $\tilde{O}(n^{0.2072})$ [21, 8, 2, 10, 23]. Despite the difficulties in improving the constant in the exponent, and as suggested by Arora [3], there is no evidence that coloring a 3-colorable graph with as few as $O(\log n)$ colors is hard.

On the other hand there are asymptotic and concrete lower bounds. Khot [25] proved that for sufficiently large k it is NP-hard to color a k-colorable graph with fewer than $k^{O(\log k)}$ colors, and this was recently improved by Huang to $2^{\sqrt[3]{k}}$ [19]. It is also known that for every constant h there exists a sufficiently large k such that coloring a k-colorable graph with hk colors is NP-hard [12]. In the non-asymptotic case, Khanna, Linial, and Safra [24] used the PCP theorem to prove it is NP-hard to 4-color a 3-colorable graph, and more generally to color a k colorable graph with at most $k + 2\lfloor k/3 \rfloor - 1$ colors. Guruswami and Khanna later gave an explicit reduction for $k = 3$ [15]. Assuming a variant of Khot's 2-to-1 conjecture, Dinur et al. prove that distinguishing between chromatic number K and K' is hard for $3 \leq K < K'$ [12]. Interestingly, this is the best conditional lower bound we give in Section 3.3, but it does not to our knowledge imply Theorem 1.

Without large strides in approximate graph coloring, we need a new avenue to approach the boundary between when coloring is difficult and when it is easy. In this paper we consider the coloring problem for a general family of graphs which we call *resiliently colorable*, in the sense that adding edges does not violate the given colorability assumption.

2 Warm up: resilient SAT

We begin by describing a resilient version of the k-satisfiability problem, which is used in proving our main result for resilient coloring in Section 4. k-SAT also serves as a nice problem to use in introducing our more general notion of resilience.

Problem (resilient k-SAT)**.** *A boolean formula φ is called* r-resilient *if it is satisfiable and remains satisfiable if any set of r variables are fixed. We call* r-resilient k-SAT *the problem of finding a satisfying assignment for an r-resiliently satisfiable formula in k-CNF form. More generally, we call r-resilient CNF-SAT the problem of finding a satisfying an assignment for an r-resilient formula in CNF form.*

While it is unclear exactly which values of k and r make this problem NP-hard, the following proposition is straightforward. We use it in Section 4 as the starting point for our main reduction.

Proposition 1. *1-resilient 6-SAT is NP-hard.*

Proof. We reduce from 3-SAT. Let $\mathbf{x} = (x_1, \dots, x_n)$ and suppose $\varphi(\mathbf{x})$ is an instance of 3-SAT. Construct $\psi = \varphi(\mathbf{x}) \vee \varphi(\mathbf{y})$ for some choice of new variables $\mathbf{y} = (y_1, \dots, y_n)$, and put ψ into 6-CNF form (which can be done in $O(n^2)$ steps). The formula φ is satisfiable if and only if ψ is, and ψ is 1-resilient: if any of the x_i are fixed, we may satisfy φ using the y_i and vice versa if any of the y_i are fixed. $\square$

Corollary 1. *For all fixed $r \geq 0$, r-resilient CNF-SAT is NP-hard.*

Proof. The technique used in the previous proof can be extended to take a disjunction of $r + 1$ copies of φ, each with different variable sets. Call this new formula ψ. Again φ is satisfiable if and only if ψ is, and we can put ψ into CNF form in polynomial time. By the pigeonhole principle if r variables are fixed then at least one disjunct has all of its variables untouched, and we can use this set of variables to satisfy the whole formula. $\square$

We now move on to graph coloring, the main subject of our study.

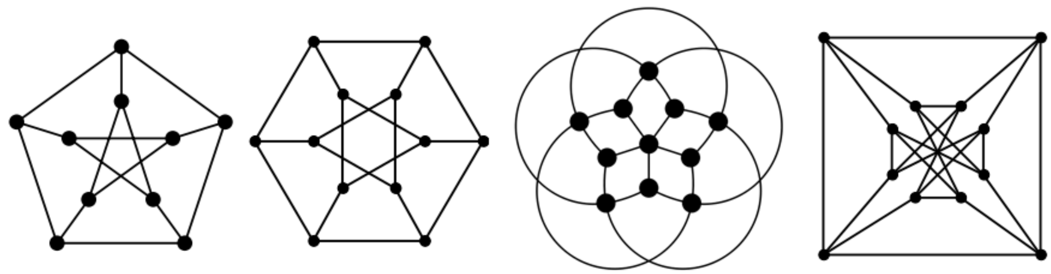


Figure 1: From left to right: the Petersen graph, 2-resiliently 3-colorable; the Dürer graph, 4-resiliently 4-colorable; the Grötzsch graph, 4-resiliently 4-colorable; and the Chvátal graph, 3-resiliently 4-colorable.

3 Resilient graph coloring, observations, and preliminary bounds

3.1 Problem definition and remarks

Problem (resilient coloring)**.** *A graph G is called* r-resiliently k-colorable *if G remains k-colorable under the addition of any set of r new edges.*[2]

We argue that this notion is not trivial by showing the resilience properties of some classic graphs. These were determined by exhaustive computer search, and are displayed in Figure 1. The Petersen graph is 2-resiliently 3-colorable. The Dürer graph is 1-resiliently 3-colorable (but not 2-resilient) and 4-resiliently 4-colorable (but not 5-resilient). The Grötzsch graph is 4-resiliently 4-colorable (but not 5-resilient). The Chvátal graph is 3-resiliently 4-colorable (but not 4-resilient).

There are a few interesting constructions to build intuition about resilient graphs. First, it is clear that every k-colorable graph is 1-resiliently $(k+1)$-colorable (just add one new color for the additional edge), but for all $k > 2$ there exist k-colorable graphs which are not 2-resiliently $(k+1)$-colorable. Simply remove two disjoint edges from the complete graph on $k+2$ vertices. A slight generalization of this argument (removing a maximal number of pairwise-disjoint edges from complete graphs) provides examples of graphs which are $\lfloor (k+1)/2 \rfloor$-colorable but not $\lfloor (k+1)/2 \rfloor$-resiliently k-colorable for all $k \geq 3$. On the other hand, every $\lfloor (k+1)/2 \rfloor$-colorable graph is $(\lfloor (k+1)/2 \rfloor - 1)$-resiliently k-colorable. This follows from the fact that r-resiliently k-colorable graphs are $(r+m)$-resiliently $(k+m)$-colorable for all $m \geq 0$ (we have one new color for each added edge).

One might expect high resilience in a k-colorable graph to be a strong enough property as to reduce the number of colors required to color it. While we expect this to be the case for super-linear resilience, there are trivial (and non-vacuous) examples of $(k-1)$-resiliently k-colorable graphs which are k-chromatic. For instance, add an isolated vertex to the complete graph on k vertices. The Grötzsch and Dürer graphs also give nice examples of k-resiliently k-colorable graphs for $k = 4$.

3.2 Observations

We are primarily interested in the complexity of coloring resilient graphs, and so we pose the question: for which values of k, r does the task of k-coloring an r-resiliently k-colorable graph admit an efficient algorithm? The following observations aid us in classifying such pairs. Using these observations and the propositions in the following section, we fill in the known complexities (including Theorem 1) in Figure 2.

Observation 1. *An r-resiliently k-colorable graph is r'-resiliently k-colorable for any $r' \leq r$. Hence, if k-coloring is in P for r-resiliently k-colorable graphs, then it is for s-resiliently k-colorable graphs for all*

[2] It is also natural to consider the removal of edges as well as their addition, but as the removal of edges can only decrease the chromatic number, it is unlikely that this form of resilience makes coloring easier.

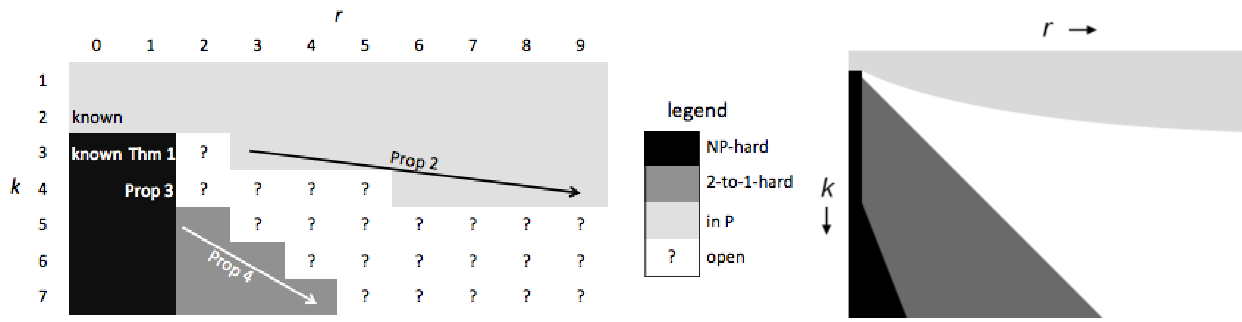


Figure 2: The known classification of the complexity of k-coloring r-resiliently k-colorable graphs. Left: the explicit classification for small k, r, referencing the propositions and theorems in this paper. Right: a zoomed-out view of the same table, with the additional NP-hard (black) region added by Proposition 5.

$s \geq r$. Conversely, if k-coloring in NP-hard for r-resiliently k-colorable graphs, then it is for s-resiliently k-colorable graphs for all $s \leq r$.

This observation manifests itself in Figure 2 by the rule that if a cell is in P, so are all of the cells to its right. Similarly, if a cell is NP-hard, so are all of the cells to its left. We have a similar observation for vertical implications.

Observation 2. *If k-coloring is in P for r-resiliently k-colorable graphs, then k'-coloring r-resiliently k'-colorable graphs is in P for all $k' \leq k$. Similarly, if k-coloring is in NP-hard for r-resiliently k-colorable graphs, then k'-coloring is NP-hard for r-resiliently k'-colorable graphs for all $k' \geq k$.*

Proof. If G is r-resiliently k-colorable, then we construct G' by adding a new vertex v with complete incidence to G. Then G' is r-resiliently $(k+1)$-colorable, and an algorithm to color G' can be used to color G. □

Observation 2 manifests itself in Figure 2 by the rule that if a cell is in P, so are all of the cells above it; if a cell is NP-hard, so are the cells below it.

More generally, we have the following observation which allows us to apply known results to our problem.

Observation 3. *If it is NP-hard to $f(k)$-color a k-colorable graph, then it is NP-hard to $f(k)$-color an $(f(k) - k)$-resiliently $f(k)$-colorable graph.*

This observation is used in Propositions 3 and 4, and follows from the fact that an r-resiliently k-colorable graph is $(r+m)$-resiliently $(k+m)$-colorable for all $m \geq 0$ (here $r = 0, m = f(k) - k$).

3.3 Upper and lower bounds

In this section we provide a simple upper bound on the complexity of coloring resilient graphs, we apply known results to show that 4-coloring a 1-resiliently 4-colorable graph is NP-hard, and we give the conditional hardness of k-coloring $(k-3)$-resiliently k-colorable graphs for all $k \geq 3$. This last result follows from the work of Dinur, et al., and depends a variant of Khot's 2-to-1 conjecture [12]. Finally, applying the result of Huang [19], we give an unconditional asymptotic lower bound. These results, along with our main theorem, are displayed graphically in Figure 2.

To explain Figure 2 more explicitly, Proposition 2 gives an upper bound for $r = \binom{k}{2}$, and Proposition 3 gives hardness of the cell $(4, 1)$ and its consequences. Proposition 4 provides the conditional lower bound, and Theorem 1 gives the hardness of the cell $(3, 1)$. Finally, Proposition 5 provides the asymptotic lower bound.

Proposition 2. *There is an efficient algorithm for k-coloring $\binom{k}{2}$-resiliently k-colorable graphs.*

Proof. If G is $\binom{k}{2}$-resiliently k-colorable, then no vertex may have degree $\geq k$. For if v is such a vertex, one may add complete incidence to any choice of k vertices in the neighborhood of v to get K_{k+1}. Finally, graphs with bounded degree $k-1$ are greedily k-colorable. $\square$

Proposition 3. *The problem of 4-coloring a 1-resiliently 4-colorable graph is NP-hard.*

Proof. It is known that 4-coloring a 3-colorable graph is NP-hard, so we may apply Observation 3. Every 3-colorable graph G is 1-resiliently 4-colorable, since if we are given a proper 3-coloring of G we may use the fourth color to properly color any new edge that is added. So an algorithm A which efficiently 4-colors 1-resiliently 4-colorable graphs can be used to 4-color a 3-colorable graph. $\square$

We call a problem *2-to-1-hard* if it is NP-hard under the condition that Khot's 2-to-1 conjecture holds.

Proposition 4. *For all $k \geq 3$, it is 2-to-1-hard to k-color a $(k-3)$-resiliently k-colorable graph.*

Proof. As with Proposition 3, we apply Observation 3 to the conditional fact that it is NP-hard to k-color a 3-colorable graph for $k > 3$. Such graphs are $(k-3)$-resiliently k-colorable. $\square$

We may apply Observation 3 once again to asymptotic bounds, such as the lower bound of Huang [19].

Proposition 5. *For sufficiently large k it is NP-hard to $2^{\sqrt[3]{k}}$-color an r-resiliently $2^{\sqrt[3]{k}}$-colorable graph for $r < 2^{\sqrt[3]{k}} - k$.*

The only unexplained cell of Figure 2 is (3,1), which we prove is NP-hard as our main theorem in the next section.

4 NP-hardness of 1-resilient 3-colorability

We now prove a non-trivial concrete lower bound, giving the hardness of 3-coloring 1-resiliently 3-colorable graphs.

Theorem 1. *It is NP-hard to 3-color a 1-resiliently 3-colorable graph.*

Proof. We reduce 1-resilient 3-coloring from 1-resilient 6-SAT. This reduction comes in the form of a graph which is 3-colorable if and only if the 6-SAT instance is satisfiable, and 1-resiliently 3-colorable when the 6-SAT instance is 1-resiliently satisfiable. We fix the three colors used in the discussion to white, black, and gray.

We first describe the gadgets involved and prove their consistency (that the 6-SAT instance is satisfiable if and only if the graph is 3-colorable), and then prove the construction is 1-resilient. Given a 6-CNF formula $\varphi = C_1 \wedge \cdots \wedge C_m$ we construct a graph G as follows. Start with a base vertex b which we may assume without loss of generality is always colored gray. For each literal we construct a *literal gadget* consisting of two vertices both adjacent to b, as in Figure 3. As such the vertices in a literal gadget may only assume the colors white and black. A variable is interpreted as true if and only if both vertices in the literal gadget have the same color. We will abbreviate this by saying a literal is *colored true* or *colored false.*

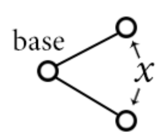
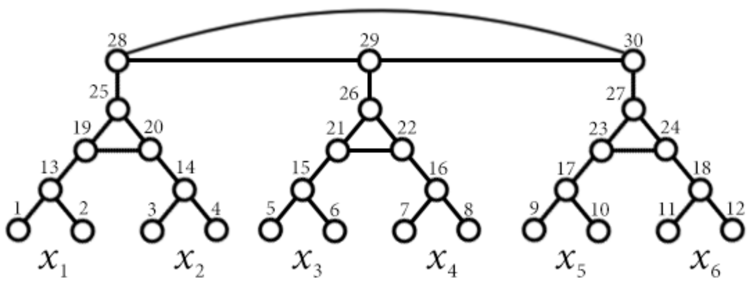


Figure 3: The gadget for a literal. The two single-degree vertices represent a single literal, and are interpreted as true if they have the same color. The base vertex is always colored gray.

We connect two literal gadgets for $x, \overline{x}$ by a *negation gadget* in such a way that the gadget for x is colored true if and only if the gadget for $\overline{x}$ is colored false. The negation gadget is given in Figure 4. In the diagram,



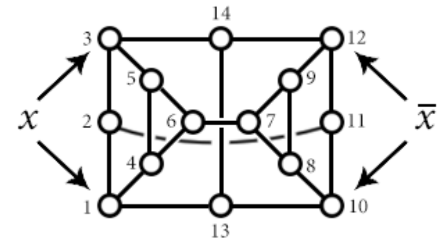


Figure 4: Left: the gadget for a clause. Right: the negation gadget ensuring two literals assume opposite truth values.

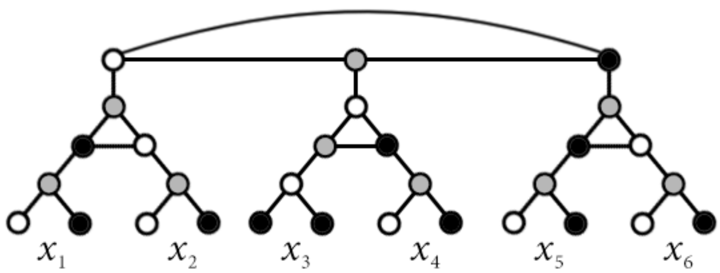


Figure 5: A valid coloring of the clause gadget when one variable (in this case x_3) is true.

the vertices labeled 1 and 3 correspond to x, and those labeled 10 and 12 correspond to $\overline{x}$. We start by showing that no proper coloring can exist if both literal gadgets are colored true. If all four of these vertices are colored white or all four are black, then vertices 6 and 7 must also have this color, and so the coloring is not proper. If one pair is colored both white and the other both black, then vertices 13 and 14 must be gray, and the coloring is again not proper. Next, we show that no proper coloring can exist if both literal gadgets are colored false. First, if vertices 1 and 10 are white and vertices 3 and 12 are black, then vertices 2 and 11 must be gray and the coloring is not proper. If instead vertices 1 and 12 are white and vertices 3 and 10 black, then again vertices 13 and 14 must be gray. This covers all possibilities up to symmetry. It is also easy to see that whenever one literal is colored true and the other false, one can extend it to a proper 3-coloring of the whole gadget.

Now suppose we have a clause involving literals, without loss of generality $x_1, \dots, x_6$. We construct the *clause gadget* shown in Figure 4, and claim that this gadget is 3-colorable if and only if at least one literal is colored true. Indeed, if the literals are all colored false, then the vertices labeled 13 through 18 in the diagram must be colored gray, and as a result the vertices 25, 26, 27 must be gray. This causes the central triangle to use only white and black, and so it cannot be a proper coloring. On the other hand, if some literal is colored true, we claim we can extend to a proper coloring of the whole gadget. Suppose without loss of generality that the literal in question is x_1, and that vertices 1 and 2 both are black. Then we see in Figure 5 how this extends to a proper coloring of the entire gadget regardless of the truth assignments of the other literals (we can always color their branches as if the literals were false).

It remains to show that G is 1-resiliently 3-colorable when φ is 1-resiliently satisfiable. This is because the worst that a new edge can do is fix the truth assignment (perhaps indirectly) of at most one literal. Since the original formula φ is 1-resiliently satisfiable, G maintains 3-colorability. Additionally, the gadgets and the representation of truth were chosen in such a way as to provide flexibility with respect to the chosen colors for each vertex, so many edges will have no effect on the colorability of G.

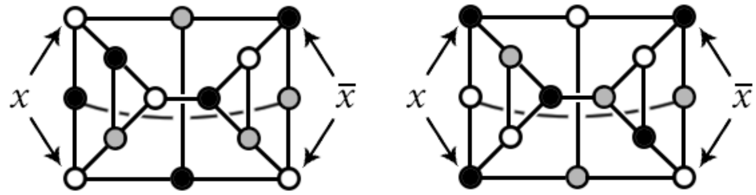


Figure 6: Two distinct ways to color a negation gadget without changing the truth values of the literals. Note that only one vertex (the rightmost center vertex) cannot be given a different color by a suitable switch between the two representations or a reflection of the graph across the horizontal axis of symmetry. If the new edge involves this vertex, we must fix the truth value appropriately.

First, one can verify that the gadgets themselves are 1-resiliently 3-colorable.[3] We will use this fact throughout the rest of the proof. We break down the analysis into eight cases based on the endpoints of the added edge. These cases are: within a single clause/negation/literal gadget, between two distinct clause/negation/literal gadgets, between clause and negation gadgets, and between negation and literal gadgets. We denote the added edge by $e = (v, w)$ and call it *good* if G is still 3-colorable after adding e.

Literal Gadgets. First, we argue that e is good if it lies within or across literal gadgets. Indeed, there is only one way to add an edge within a literal gadget, and this has the effect of setting the literal to false. If e lies across two gadgets then it has no effect: if c is a proper coloring of G without e, then after adding e either c is still a proper coloring or we can switch to a different representation of the truth value of v or w to make e properly colored (i.e. swap "white white" with "black black," or "white black" with "black white" and recolor appropriately).

Negation Gadgets. Next we argue that e is good if it involves a negation gadget. Let N be a negation gadget for the variable x. Indeed, by 1-resilience an edge within N is good; e only has a local effect within negation gadgets, and it may result in fixing the truth value of x. Now suppose e has only one vertex v in N. Figure 6 shows two ways to color N, which together with reflections along the horizontal axis of symmetry have the property that we may choose from at least two colors for any vertex we wish. That is, if we are willing to fix the truth value of x, then we may choose between one of two colors for v so that e is properly colored regardless of which color is adjacent to it.

Clause Gadgets. It remains to inspect the case that e lies within a clause gadget or between two clause gadgets. As with the negation gadget, it suffices to fix the truth value of one variable suitably so that one may choose either of two colors for one end of the new edge. Although it is not always necessary to fix a variable to accommodate the new edges, it is possible in all cases. Figure 7 provides a detailed illustration of one such case. Here, we focus on two branches of two separate clause gadgets, and add the new edge $e = (v, w)$. The added edge has the following effect: if x is false, then neither y nor z may be used to satisfy C_2 (as w cannot be gray). This is no stronger than requiring that either x be true or y and z both be false, i.e., we add the clause $x \vee (\overline{y} \wedge \overline{z})$ to φ. Since this clause can be satisfied by fixing a single variable (x to true), and φ is 1-resilient, we can still satisfy φ and hence 3-color G. The other cases are nearly identical, except that the added clause changes; fixing a single variable suffices to satisfy the new clause.

This proves that G is 1-resilient when φ is, and finishes the proof. □

We note that the clause gadget in the proof above comes from Kun et al. [29], who employ this construction in analyzing a different graph coloring problem.

[3] Again, these graphs are sufficiently small so as to admit verification by computer search.

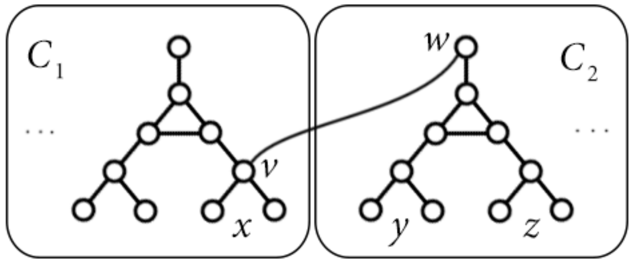


Figure 7: An example of an edge added between two clauses C_1, C_2.

5 Discussion and open problems

The notion of resilience introduced in this paper leaves many questions unanswered, both specific problems about graph coloring and more general exploration of resilience in other combinatorial problems and CSPs.

Regarding resilient graph coloring, our paper established the fact that 1-resilience doesn't affect the difficulty of graph coloring. However, the question of 2-resilience is wide open, as is establishing linear lower bounds without dependence on the 2-to-1 conjecture. There is also obvious room for improvement in finding efficient algorithms for highly-resilient instances, closing the gap between NP-hardness and tractability.

Additionally, we only scratched the surface of questions about resilient satisfiability. For instance, while 3-resilient 3-SAT is vacuously trivial, it is unclear whether 1- or 2-resilience is strong enough to make finding a satisfying assignment of a 3-SAT instance easy. We conjecture that 1-resilience is NP-hard and 2-resilience is tractable.

On the general side, there is a wealth of NP-complete problems that our framework could be applied to, including Hamiltonian circuit, set cover, 3D-matching, 0-1 linear programming, and many others. Each presents its own boundary between NP-hardness and tractability, and there are undoubtedly interesting relationships across problems, as we showed with resilient satisfiability and resilient coloring.

Acknowledgments

We thank Shai Ben-David for helpful discussions.

References

[1] Margareta Ackerman and Shai Ben-David. Clusterability: A theoretical study. *Journal of Machine Learning Research - Proceedings Track*, 5:1–8, 2009.

[2] Sanjeev Arora and Eden Chlamtac. New approximation guarantee for chromatic number. In *STOC*, pages 215–224, 2006.

[3] Sanjeev Arora and Rong Ge. New tools for graph coloring. In *APPROX-RANDOM*, pages 1–12, 2011.

[4] Pranjal Awasthi, Avrim Blum, and Or Sheffet. Center-based clustering under perturbation stability. *Inf. Process. Lett.*, 112(1-2):49–54, 2012.

[5] Bonnie Berger and John Rompel. A better performance guarantee for approximate graph coloring. *Algorithmica*, 5(3):459–466, 1990.

[6] Yonatan Bilu and Nathan Linial. Are stable instances easy? *Combinatorics, Probability & Computing*, 21(5):643–660, 2012.

[7] Avrim Blum. New approximation algorithms for graph coloring. *J. ACM*, 41(3):470–516, 1994.

[8] Avrim Blum and David R. Karger. An $\tilde{O}(n^{3/14})$-coloring algorithm for 3-colorable graphs. *Inf. Process. Lett.*, 61(1):49–53, 1997.

[9] Leizhen Cai. Parameterized complexity of vertex colouring. *Discrete Applied Mathematics*, 127(3):415–429, 2003.

[10] Eden Chlamtac. *Non-local analysis of SDP-based approximation algorithms.* PhD thesis, Princeton University, Princeton, NJ, USA, 2009. AAI3338674.

[11] David P. Dailey. Uniqueness of colorability and colorability of planar 4-regular graphs are np-complete. *Discrete Mathematics*, 30(3):289 – 293, 1980.

[12] Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring. *SIAM J. Comput.*, 39(3):843–873, 2009.

[13] David Eppstein, Marshall W. Bern, and Brad L. Hutchings. Algorithms for coloring quadtrees. *Algorithmica*, 32(1):87–94, 2002.

[14] Michael R. Garey and David S. Johnson. The complexity of near-optimal graph coloring. *J. ACM*, 23(1):43–49, 1976.

[15] Venkatesan Guruswami and Sanjeev Khanna. On the hardness of 4-coloring a 3-colorable graph. *SIAM J. Discrete Math.*, 18(1):30–40, 2004.

[16] Magnús M. Halldórsson. A still better performance guarantee for approximate graph coloring. *Inf. Process. Lett.*, 45(1):19–23, 1993.

[17] Johan Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[18] Chính T. Hoàng, Frédéric Maffray, and Meriem Mechebbek. A characterization of b-perfect graphs. *Journal of Graph Theory*, 71(1):95–122, 2012.

[19] Sangxia Huang. Improved hardness of approximating chromatic number. *CoRR*, abs/1301.5216, 2013.

[20] David S. Johnson. Worst case behavior of graph coloring algorithms. In *Proc. 5th Southeastern Conf. on Comb., Graph Theory and Computing*, pages 513–527, 1974.

[21] David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, 1998.

[22] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.

[23] Ken-ichi Kawarabayashi and Mikkel Thorup. Combinatorial coloring of 3-colorable graphs. In *FOCS*, pages 68–75, 2012.

[24] Sanjeev Khanna, Nathan Linial, and Shmuel Safra. On the hardness of approximating the chromatic number. *Combinatorica*, 20(3):393–415, 2000.

[25] Subhash Khot. Improved inaproximability results for maxclique, chromatic number and approximate graph coloring. In *FOCS*, pages 600–609, 2001.

[26] Daniel Kobler and Udi Rotics. Edge dominating set and colorings on graphs with fixed clique-width. *Discrete Applied Mathematics*, 126(2-3):197–221, 2003.

[27] Daniel Král, Jan Kratochvíl, Zsolt Tuza, and Gerhard J. Woeginger. Complexity of coloring graphs without forbidden induced subgraphs. In *WG*, pages 254–262, 2001.

[28] Vipin Kumar. Algorithms for constraint-satisfaction problems: A survey. *AI magazine*, 13(1):32, 1992.

[29] Jeremy Kun, Brian Powers, and Lev Reyzin. Anti-coordination games and stable graph colorings. In *SAGT*, pages 122–133, 2013.

[30] Matús Mihalák, Marcel Schöngens, Rastislav Srámek, and Peter Widmayer. On the complexity of the metric tsp under stability considerations. In *SOFSEM*, pages 382–393, 2011.

[31] Lev Reyzin. Data stability in clustering: A closer look. In *ALT*, pages 184–198, 2012.

[32] Avi Wigderson. Improving the performance guarantee for approximate graph coloring. *J. ACM*, 30(4):729–735, 1983.

[33] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007.